INF

Studiengang
Medien- und
Kommunikationsinformatik

Hochschule Reutlingen
Reutlingen University

Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

Informatics Inside:

Human-Centered Computing

Informatik-Konferenz an der Hochschule Reutlingen
30. April 2014

ISBN 978-3-00-045427-1

Impressum

Anschrift:
Hochschule Reutlingen
Reutlingen University
Fakultät Informatik
Medien- und Kommunikationsinformatik
Alteburgstraße 150
D-72762 Reutlingen

Telefon: +49 7121 / 271-4002
Telefax: +49 7121 / 271-4042

E-Mail: infoinside@reutlingen-university.de
Internet: http://www.infoinside.reutlingen-university.de

Hochschule Reutlingen
Reutlingen University

Herstellung und Verlag: Hochschule Reutlingen
ISBN 978-3-00-045427-1

Inhaltsverzeichnis

Gestenerkennung & Augumented Virtuality

Thomas Bauer
Anforderungsanalyse zur computergestützten Erkennung der Deutschen Gebärdensprache *8*

Matthias Gutekunst
Augmented Virtuality zur Steigerung der Immersion in virtuellen Umgebungen *26*

Stanislas Mauser
Analysis of Finger- and Palm-based interaction paradigms for Touch-Free Gesture-Based Control of Medical Devices with the Leap Motion Controller *34*

Softwaretechnik

René Mangold
Selektion von Szenarien zur Optimierung von Simulationen im präventiven Krisenmanagement *46*

Arkadius Weister
Language Oriented Programming: Modulare domänenspezifische Sprachen *54*

Entwicklung Mobiler Anwendungen

Olaya De la Rosa Avitia
Strategy to Test Mobile Apps *70*

Viktoria Hoffmann
Optimierung der Usability von digitalen Fahrtenbüchern durch automatisches Erfassen von fahrzeugspezifischen Daten *80*

Johannes Kartheininger
Vergleich der Single Sign On Verfahren SAML und OpenID Connect *92*

Virtuelle Welten

André Antakli
Umgebungswahrnehmung von agentenbasierten simulierten Menschmodellen in virtuellen Welten im Kontext C3D *100*

Vergleich der Single Sign On Verfahren SAML und OpenID Connect

Johannes Kartheininger
Reutlingen University
Johannes.Kartheininger@Student.Reutlingen-University.de

Abstract
In dieser Arbeit werden die beiden Single Sing On-Protokolle SAML und OpenID Connect miteinander verglichen. Vorab wird die Aufgabe eines Single Sign On-Systemes erläutert und allgemeine Vorteile sowie Nachteile aufgezählt.

Schlüsselwörter
Authentifikation, Autorisation, Single Sign On, Identitätsmanagment

CR-Kategorien
D.4.6 [Security and Protection]: Authentication; C.2.m [Computer Systems Organization]: Computer- Communication Networks – Miscellaneous

1 Einleitung
In Unternehmen oder öffentlichen Institutionen ist es bereits der Fall, dass der Nutzer um Zugriff auf für ihn relevante Anwendungen und Ressourcen zu erhalten, sich nur einmal authentifizieren muss. Bei öffentlichen Anwendungen im Internet ist dieser Trend nun ebenso zu erkennen. So prangern auf vielen Seiten bereits verschiedene Loginverfahren, bei denen der Nutzer sich mit einem bereits verbundenen Dienst gegenüber einem neuen Dienst identifizieren kann (siehe exemplarisch Abbildung 1).

Abbildung 1 Loginmöglichkeiten auf StackOverflow.com

Dies erspart ihm sowohl eine aufwendige Registrierung bei dem neu zu nutzenden Dienst, als auch das Verwalten mehrerer Passwörter. Man nennt dieses Verfahren auch Single Sign On.

2 Motivation
Durch die steigende Verbreitung vieler verteilter Einzelsysteme, die dennoch zusammenarbeiten, besteht der Bedarf die Authentifizierung des Nutzers nicht unnötig oft zu verlangen. Es soll dem Nutzer ermöglicht werden sich einmal gegenüber einem System zu authentifizieren und danach alle für ihn relevanten Systeme nutzen zu können. Single Sign On soll die Anzahl der Loginvorgänge reduzieren, sowie die Sicherheit und Effizienz erhöhen und die Usability zu verbessern (vgl. [6] & [7], S.10f).

Ebenso soll die Interoperabilität der IT-Landschaft gesteigert werden, indem die

---

Betreuer Hochschule: Prof. Dr.-Ing. habil. Natividad Martínez Madrid
Hochschule Reutlingen
natividad.martinez@reutlingen-university.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen

Authentifizierung nicht mehr dezentral an jedem System ausgeführt sowie seitens der Administration verwaltet werden muss. Durch Single Sign On reduzieren sich ebenso Datenredundanzen (vgl. [8], S.1f).

Obwohl es Single Sign On bereits seit längerer Zeit gibt (bspw. Kerberos [1] seit 1980), so gibt es durch die wachsende Cloudlandschaft, sowie das Aufkommen mobiler Geräte wie Smartphones und Tablets neue Anforderungen an Single Sign On-Systeme. So wurde erst kürzlich ein neuer Standard im Bereich Single Sign On verabschiedet, welcher eben jene Anforderungen der mobilen Geräte erfüllt (siehe [15]).

3 Definitionen

Um die Aufgabe eines Single Sign On Verfahrens nachvollziehen zu können, ist es wichtig die Unterschiede der Begriffe Autorisation und Authentifizierung zu kennen.

Authentifizierung

Die Authentifizierung bezeichnet im Kontext von Single Sign On den Nachweis eines Nutzers, dass er derjenige ist, für den er sich ausgibt.

Autorisation

Die Autorisation bezeichnet die Erlaubnis eines Nutzers eine gewisse Aktion auszuführen, bspw. auf eine bestimmte Ressource zuzugreifen oder Zugang zu einem System zu erhalten.

Single Sign On

Single Sign On (SSO) bezeichnet die einmalige Authentifizierung eines Nutzers gegenüber einem einzelnen System, woraufhin er Zugriff auf alle Systeme erhält für die er autorisiert ist. Eine Kerneigenschaft von Single Sign On-Systemen ist es, dass die Authentifizierung nicht an jedem System nochmals stattfinden muss. Die Überprüfung der Autorisation übernimmt jedoch nicht zwingend das SSO-System, sondern das zu nutzende System kann selbst überprüfen ob der authentifizierte Nutzer dazu berechtigt ist. Ebenso gibt es jedoch auch Systeme die beides kombiniert anbieten oder nur die Autorisierung übernehmen.

4 Single Sign On

SSO soll sowohl auf Seiten der Nutzer, als auch auf Seiten der Administration einiges vereinfachen. Aus Nutzersicht wären dies bspw. die Verwaltung weniger Konten sowie eines einzigen Logins und aus administrativer Sicht wird bspw. die Anzahl an Serviceanfragen bzgl. neu zu setzender Passwörter bei vergessenen Zugangsdaten geringer werden.

4.1 Vorteile

Theoretisch bedeutet für den Nutzer, dass wenn er nur einen Account zu verwalten hat, ein sicheres Passwort ausreicht. Untersuchungen ergaben, dass Nutzer dazu neigen Passwörter bei verschiedenen Diensten mehrmals zu verwenden, sowie einfach zu erratende Kombinationen nutzen (vgl. [11], S.26f, S.34-37). Ebenso entfällt für den Nutzer das Registrieren bei jeder einzelnen Anwendung.

Aus Sicht eines Unternehmens welches ein Single Sign On-Verfahren für ihre Anwendungen anbietet, verringert sich der Verwaltungsaufwand im Bezug auf Nutzerkonten. Es muss nur eine zentrale Identitätskomponente administriert werden, anstatt mehrere verschiedene. Ebenso benötigen neue Systeme, die nachträglich hinzugefügt werden, keine eigene Identitätskomponente, sondern können auf die bereits vorhandene zurückgreifen.

4.2 Nachteile

Es gibt jedoch auch einige negative Aspekte, die man sowohl bei der Entwicklung eines eigenen Single Sign On – Systems als auch bei der Auswahl eines fertigen Single Sign On – Systems bedenken muss. So kann der Ausfall der einzelnen Authentifikationskomponente bedeuten, dass die Nutzer ihre gesamten anderen Systeme während des Ausfalls nicht mehr verwenden können. Man könnte auch sagen, dass dies ein Single Point of Failure darstellen kann. Dies bedeutet, dass

die Authentifikations-komponente sowohl skalierbar sein muss, als auch ein Backupmechanismus/-system vorhanden sein muss, der die Verwendbarkeit der anderen Systeme ermöglicht.

Sollte das SSO-System Sicherheitslücken aufweisen, so sind alle Systeme betroffen, die auf dieses SSO-System als Identitäts-lieferanten zurückgreifen.

5 Single Sign On-Systeme

Single Sign On-Systeme werden unterschiedlich eingeordnet. In dieser Ausarbeitung wird die Unterscheidung von Rhada und Clerq angewandt (siehe [12], S.134f & [13], S. 44f). In Abbildung 2 ist diese Unterscheidung skizziert, sowie die hier betrachteten Web Single Sign On Systeme eingezeichnet.

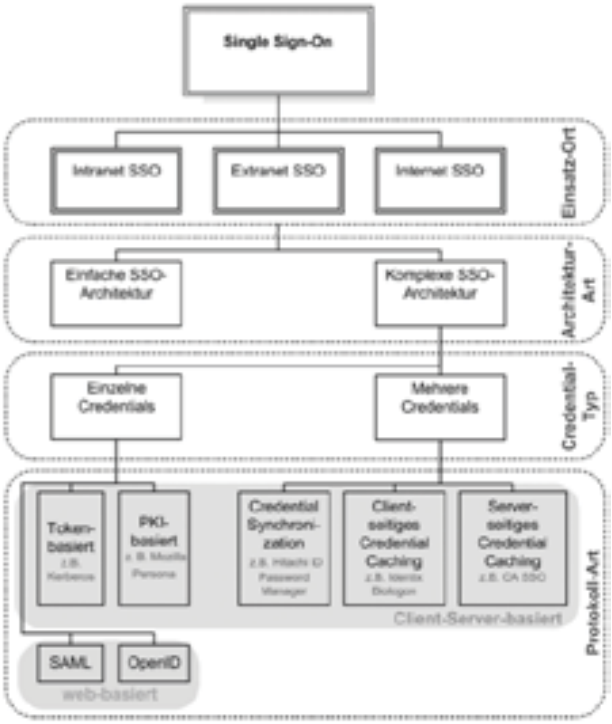

Abbildung 2 Kategorisierung von Single Sign On-Systemen ([11], S.58)

5.1 SAML 2.0

Security Assertion Markup Language 2.0 (SAML 2.0) ist ein auf XML basierendes Rahmenwerk für die Beschreibung von Autorisations- sowie Authentifikationsdaten und deren Austausch zwischen verschiedenen Systemen (siehe [2]).

In einem Single Sign On-System, welches SAML verwendet, trifft man die Rollen Identity Provider (IDP) sowie Service Provider (SP) an. Ein SP ist dabei eine Webanwendung, ein Webservice usw., welcher von einem Nutzer oder einem System genutzt werden möchte. Bevor ein SP genutzt werden kann, muss die Identität des Nutzers von einem IDP festgestellt werden. Ob dieser Nutzer nun den Service verwenden kann, obliegt dem SP, der IDP betrifft einzig und allein die Authentifizierung, nicht die Autorisierung. Die Kommunikation zwischen SP und IDP erfolgt üblicherweise über den Client, d.h. es findet keine direkte Kommunikation zwischen den beiden Parteien statt. Folgendes Schaubild stellt schemenhaft die verschiedenen Rollen in einem SSO-Szenario vor, sowie deren Beziehungen untereinander.

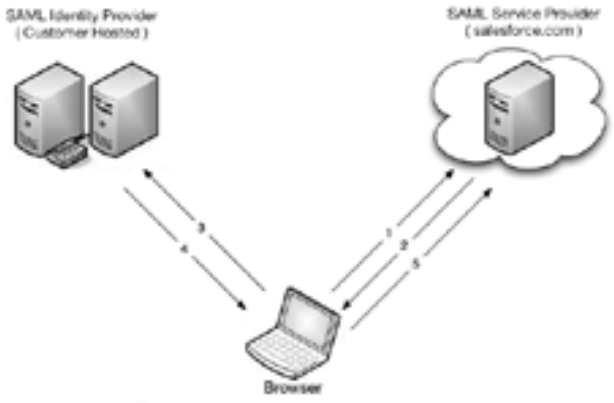

Abbildung 3 SAML Schaubild[1]

5.2 OAuth 2.0

OAuth 2.0 ist ein offener Standard für SSO der nur die Autorisierung behandelt (siehe RFC6749[2]). So ist es möglich bei Anwendungen über OAuth 2.0 auf Ressourcen anderer Anwendungen zuzugreifen. Eine Authentifizierung des Nutzers findet jedoch nicht explizit statt, kann jedoch implizit erfolgen (siehe Abschnitt Authentifizierung & Autorisierung).

---

[1] http://wiki.developerforce.com/page/File:Saml_flow.png

[2] http://tools.ietf.org/html/rfc6749

OAuth 2.0 definiert vier Rollen[3]:

- resource owner
- resource server
- client
- authorization server

In Abbildung 4 sind die Beziehungen der Rollen untereinander dargestellt. Der User entspricht hierbei der Rolle des „resource owner“.

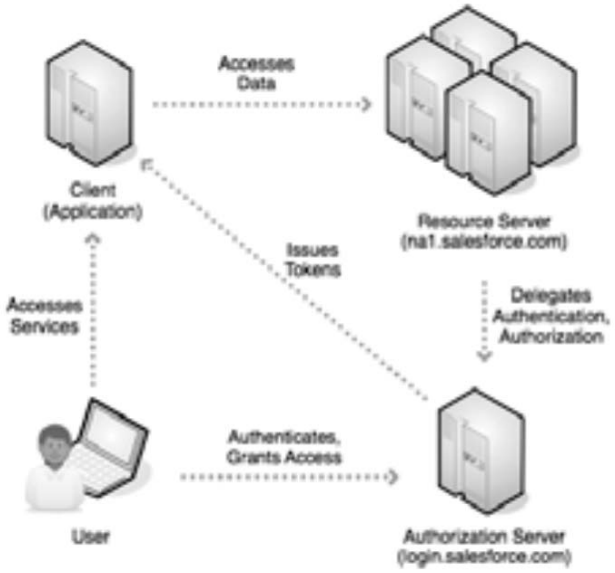

Abbildung 4 OAuth Schaubild[4]

5.2.1 OpenID Connect

Ein im Februar 2014[5] erschienener Standard für Single Sign On basiert auf OAuth 2.0 und erweitert diesen um eine Authentifizierungskomponente. Bereits kurz nachdem der finale Standard verabschiedet wurde, gibt es OpenID Connect-Anbieter, die bereits während der Entwurfsphase des Standards zuerst hybride Mechanismen zur Verknüpfung von OAuth 2.0 und dem ursprünglichen OpenID anboten und nun bereits OpenID Connect implementiert haben (bspw. Google[6] und Paypal[7]). Für einen Aufbau von OpenID Connect siehe Abbildung 4, da die Rollen, sowie Abfolgen exakt wie bei OAuth 2.0 definiert sind. Bei OpenID Connect ist es nur zusätzlich möglich Identitätsinformationen vom Authorization Server zu erhalten.

6 Detailbetrachtung

Auch wenn der Einsatzzweck von SAML 2.0 sowie OpenID Connect ein ähnlicher ist, unterscheiden sich beide Protokolle auf technischer Ebene.

6.1 Authentifizierung & Autorisierung

Ist SAML sowohl für Authentifizierung als auch Autorisierung verwendbar, so übernimmt OAuth 2.0 einzig die Autorisierung. Es kann implizit eine Autorisierung stattfinden, da nur der Besitzer eines Tokens auch Zugriff auf diesen hat. Es kann also bspw. der Token für den Zugriff auf irgendeine Komponente angefordert werden und beim nächsten Authentifizierungsvorgang überprüft werden ob immer noch der Nutzer der Zugriff auf diese Komponente hat. OAuth ist aber eigentlich eine reine Autorisierungskomponente und davon OAuth 2.0 für Authentifizierung zu verwenden wird abgeraten[8]. Es kann nicht davon ausgegangen werden, dass der Zugang zu einer Ressource gleichzeitig bedeutet, dass nur ein Nutzer eben Zugriff auf jene Ressource erhält. In einzelnen Fällen wie bspw. dem Zugriff auf Kontoinformationen von Facebook mag dies zutreffen. Jedoch sagt bspw. der Zugriff auf ein Dokument in GoogleDrive nichts über die

[3] http://tools.ietf.org/html/rfc6749#section-1.1

[4] https://wiki.developerforce.com/page/File:OAuthRoles.png

[5] http://openid.net/2014/02/26/the-openid-foundation-launches-the-openid-connect-standard/

[6] https://developers.google.com/accounts/docs/OAuth2Login

[7] https://developer.paypal.com/docs/integration/direct/identity/log-in-with-paypal/

[8] http://www.threadsafe.com/2012/01/problem-with- oauth-forauthentication.html

Identität des Nutzers aus, da mehrere Nutzer Zugriff darauf haben könnten.

Aufbauend auf dem Konzept, dass OAuth 2.0 einzig zur Autorisierung verwendet werden sollte, wurde OpenID Connect entwickelt um eine Möglichkeit zu haben Nutzer auch authentifizieren zu können.

6.2 Token/Nachrichtenformat

SAML sieht es vor, dass die Informationen in einem XML-Format ausgetauscht werden.

Bei OpenID Connect können die Informationen jedoch binär, per JavaScript Object Notation (JSON) oder gar per SAML-Format (Ein im SAML-Standard deklariertes XML-Format) übertragen werden.

6.3 Transport

Sieht OpenID Connect nur vor Tokens per Http(s) zu übertragen, so gibt es für SAML ebenso Bindings welche eine Übertragung per Http(s) vorsehen, jedoch ist auch bspw. SOAP möglich. Ebenso steht es dem Entwickler offen auch bspw. Java Message Service zu verwenden, da bei SAML eben keine festen Vorgaben gemacht wurden.

6.4 Single Logout

Single Logout bezeichnet den umgekehrten Vorgang zu Single Sign On. Wird eine Single Sign On-Session nicht geschlossen, ist der Zugang zu allen anderen Services die auf diesen SSO-Provider zugreifen möglich. Wird also bspw. an einem öffentlichen PC die Session nicht geschlossen oder die Session per Session Hijacking[9] von einem Angreifer übernommen, so ist sowohl der Zugang zu dem System gegenüber dessen man sich identifiziert hat möglich, als auch alle anderen Systeme, die auf diesen SSO-Provider zurückgreifen.

Ist es bspw. möglich in SSO-Systemen die auf SAML setzen sich zentral abzumelden, so sieht der OpenID Connect Standard nur vor, dass nachdem auf einem einzelnen System ein Logout getätigt wurde, der Nutzer abermals gefragt wird, ob er die gesamte Session beenden möchte[10]. Dies ist jedoch nur durch Verwendung eines Inlineframes (Einbindung einer anderen Webseite/-inhaltes in einem vordefinierten Bereich) möglich und dadurch ebenso nur in Webbrowsern und nicht in nativen Anwendungen auf mobilen Geräten. Ebenso ist die Spezifikation zu Session-Managment immer noch in einem Entwurfsstadium, obwohl der OpenID Connect-Standard bereits verabschiedet wurde[11].

6.5 Anwendungsbereiche

Beide SSO-Varianten können sowohl in Unternehmen, als auch im öffentlichen Internet verwendet werden. Dennoch kann festgehalten werden, dass SAML eher innerhalb bzw. zwischen Unternehmen Anklang findet und im Web eher OAuth 2.0 oder OpenID Connect zu finden ist (vgl. [14], S.5f).

6.5.1 Mobile

OpenID Connect wurde mit dem Gedanken entworfen ein einfacher und auf allen Geräten funktionierende Single Sign On-Standard zu sein. Erreicht wird dies dadurch, dass das Tokenformat per JSON ausgegeben werden kann um den Overhead bei XML zu sparen, aber auch dadurch, dass die Authentifizierung nicht zwingend per Webbrowser stattfinden muss. Im Gegensatz dazu ist bei SAML vorgesehen, dass die Authentifizierung des Nutzers über einen Webbrowser erfolgt. Auf mobilen Geräten jedoch können Anwendungen auch ausserhalb des Webbrowsers ausgeführt werden.

6.6 Sicherheit

Ist Verschlüsselung bei SAML bereits auf Nachrichtenebene geläufig und auf Transportebene vorgeschrieben, so wird bei OpenID Connect und OAuth 2.0 die

[9] http://projects.webappsec.org/w/page/13246944/Insufficient Session Expiration

[10] http://openid.net/specs/openid-connect-session-1_0.html#RPLogout

[11] http://openid.net/developers/specs/

Verschlüsselung auf Nachrichtenebene selbst optional deklariert. Andererseits ist bei OpenID Connect die Verschlüsselung auf Transportebene ebenso zwingend vorgesehen (siehe [12] S.10f).

7 Zusammenfassung

Ist SSO zwar ein Thema für das es bereits alte und bewährte Lösungen gibt, ist es jedoch nötig aufgrund neuartiger Geräte, sowie bspw. dem Wechsel zwischen Internet-verbindungen (Mobilfunk und Wlan), die bewährten Techniken anzupassen oder nochmals neu zu erarbeiten (siehe neu erarbeitetes OpenID Connect**Fehler! Textmarke nicht definiert.**). Ebenso muss bei Einsatz eines Single Sign On-Systemes je nach Kontext nicht nur die technische Seite betrachtet werden, sondern auch die psychologische (siehe [9], S.65f & [10]). Von dieser Betrachtungsweise aus kann eine Ablehnung einer zentralisierten Authen-tifizierungsstelle psychologische Gründe haben, so möchten Nutzer explizit von-einander unabhängige Pseudoidentitäten um so bspw. privat und geschäftlich strikt voneinander zu trennen. Auch gilt es je nachdem ob SSO innerhalb eines Unter-nehmens oder im öffentlichen Internet be-nötigt wird, abzuwägen welche SSO-Technik verwendet wird. Selbst Lösungen die einst für SSO in beiden Bereichen geeignet waren, sollten nach einer Überarbeitung genauer analysiert werden. So ist bspw. OAuth 2.0 im Gegensatz zu OAuth 1.0 in der Standard-umsetzung weniger sicher, da die Ver-schlüsselung der Authentifizierungsinforma-tionen entfernt wurde und das ganze Ver-fahren nur noch auf Transportebene über TLS verschlüsselt wird (siehe Austritts-statement einer der Mitbegründer von OAuth 1.0 sowie Anfangs auch OAuth 2.0[13]). Einer der Hauptentscheidungsgründe für eine der beiden Varianten wird die Integration in die bestehende IT-Landschaft sein, sowie die zukünftige Erweiterung der selbigen. Ist es beispielsweise angedacht Webanwendungen auch per mobilen Endgeräten wie Tablets und Smartphones erreichbar zu machen, so könnte die Entscheidung eher zu OpenID Connect tendieren, da hierbei direkt das vom SSO-Server generierte Zugangstoken als JSON vorliegt und dadurch eine einfachere Verarbeitung auf den Geräten stattfinden kann. SAML benötigt anfangs mehr Konfigurationsaufwand, gilt jedoch als altbe-währtes Verfahren als sicher. OpenID Connect muss erst noch ausreichender betrachtet werden um Aussagen bezüglich Sicherheit treffen zu können.

[12] https://blog.surfnet.nl/wp-content/uploads/2013/04/SURFnet-OpenID-Connect-1.1-.pdf

[13] http://hueniverse.com/2012/07/oauth-2-0-and-the-road-to-hell/

8 Literaturverzeichnis

[1] Kerberos: The Network Authentication Protocol. Website, 2014. Online verfügbar unter http://web.mit.edu/kerberos/; Besucht am 10.01.2014.

[2] OASIS Security Services (SAML) TC. Website, 2014. Online verfügbar unter https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security; Besucht am 10.01.2014

[3] Shibboleth. Website, 2014. Online verfügbar unter https://shibboleth.net/; Besucht am 10.01.2014

[4] OAuth 2.0. Website, 2014. Online verfügbar unter http://oauth.net/2/; Besucht am 10.01.2014

[5] OpenID Connect. Website, 2014. Online verfügbar unter http://openid.net/connect/; Besucht am 10.01.2014

[6] Revar, A.G.; Bhavsar, M.D., "Securing user authentication using single sign-on in Cloud Computing," Engineering (NUiCONE), 2011 Nirma University International Conference on , vol., no., pp.1,4, 8-10 Dec. 2011

[7] S. Xiong. “Web Single Sign-On System For WRL Company”, Royal Institute of Technology (KTH) Stockholm, Sweden, 2005

[8] D. Stojceski. Konzeption einer Kerberos-basierten Single Sign-On Lösung für ein ausgewähltes Szenario im Hochschulbereich. Fachhochschule Bonn-Rhein-Sieg, Fachbereich Informatik, Bachelorthesis, Bonn-Rhein-Sieg, 2006

[9] San-Tsai Sun, Yazan Boshmaf, Kirstie Hawkey, and Konstantin Beznosov. 2010. A billion keys, but few locks: the crisis of web single sign-on. In Proceedings of the 2010 workshop on New security paradigms (NSPW '10). ACM, New York, NY, USA, 61-72. 2010

[10] San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov. What makes users refuse web single sign-on? an empirical investigation of OpenID. In Proceedings of Symposium on Usable Privacy and Security, July 2011

[11] S. Langer. Sicherheit von passwortbasierten Authentifizierungssystemen. Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik & Informatik, Bachelorthesis, Hamburg. 2013

[12] V. Radha, D.H. Reddy. A Survey on Single Sign-On Techniques. In: Procedia Technology 4 (2012), 134–139. Hyderabad, Indien. 2012

[13] J. Clerq. Single Sign-On Architectures. In Infrastructure Security: Lecture Notes in Computer Science Volume 2437, 2002, pp 40-58. 2

[14] Ping Identity. A Standards-based Mobile Application IdM Architecture. Whitepaper, 2012

[15] OpenID Connect Pressemitteilung. Website, 2014. Online verfügbar unter http://openid.net/2014/02/26/the-openid-foundation-launches-the-openid-connect-standard/; Besucht am 10.01.2014